



# Playing with geolocation data

Sameh Ben Fredj  
XKE April 2019

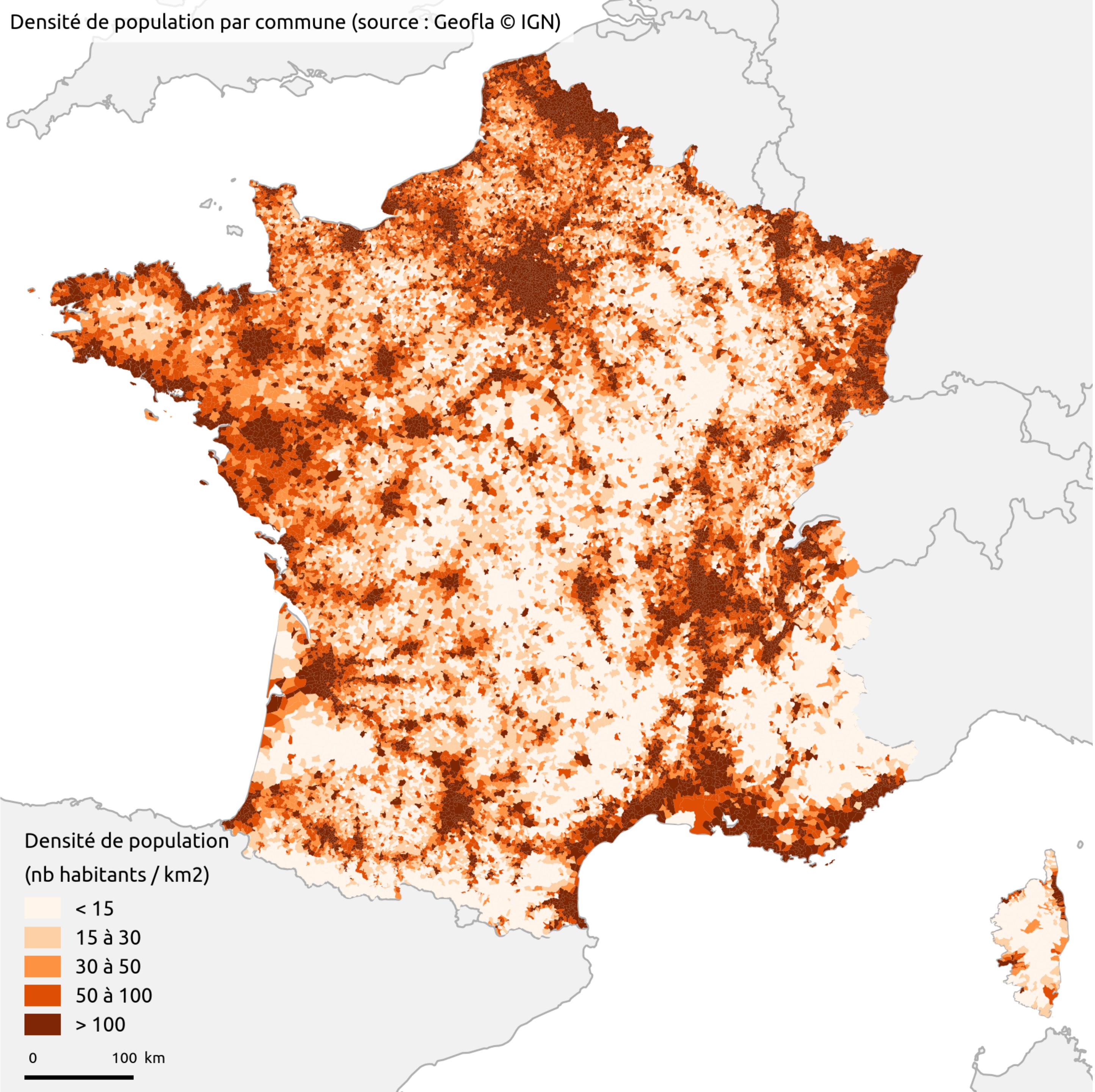
# Agenda

1. Why geo data ? (5')
2. Folium API + Quick Warm Up (10')
3. Hands on with Folium (30')
4. Going further (5')

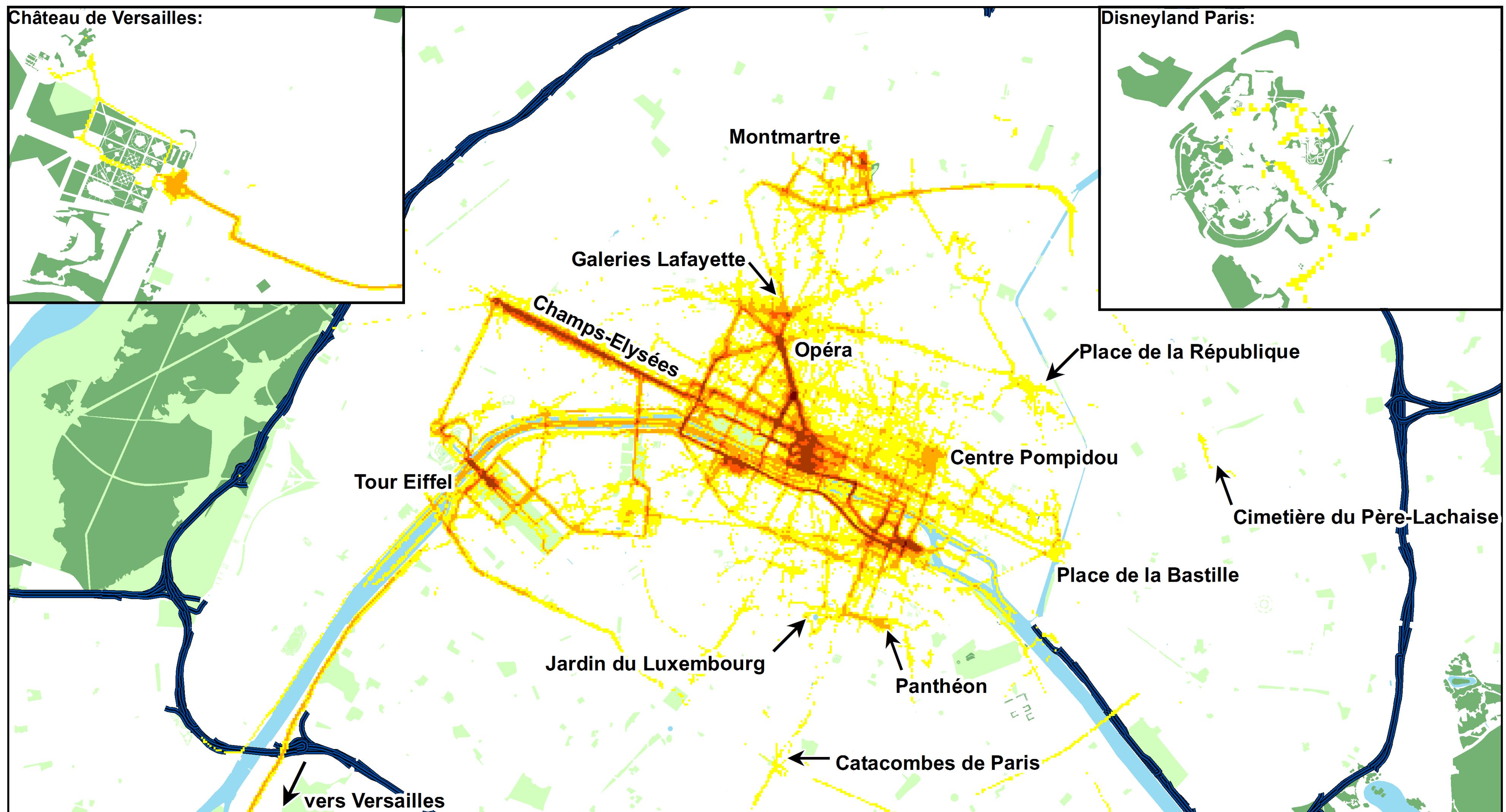
# **Why Geo Data ?**

**« Any data having connection or relation to any location on earth »**

Densité de population par commune (source : Geofla © IGN)

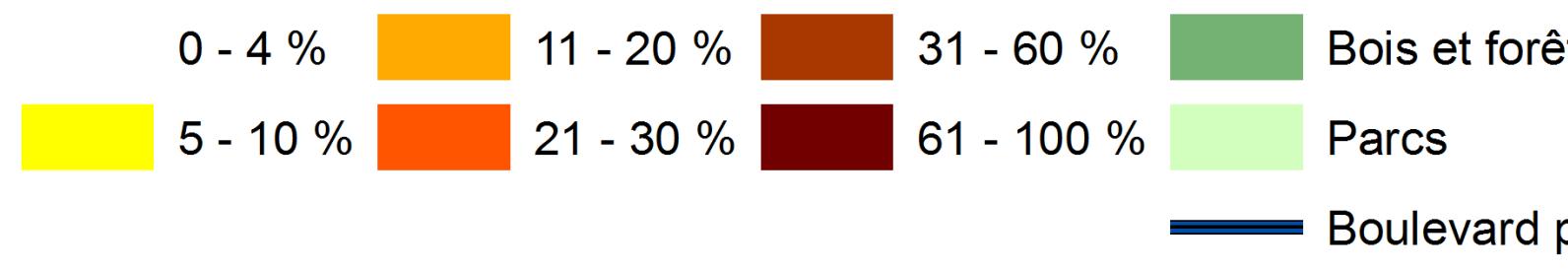






### Modèles de mobilité touristique à Paris (n=129)

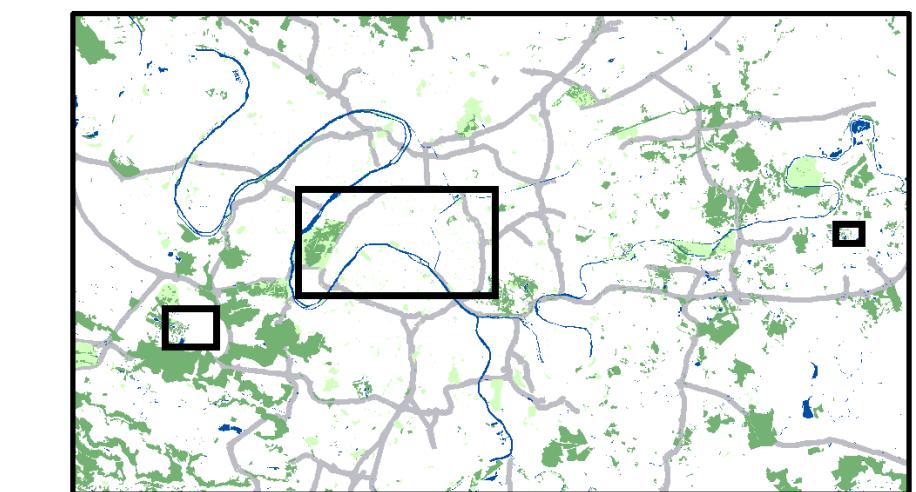
Taux de fréquentation par rapport à la cellule la plus fréquentée (pixel de 20 m), arrondi au nombre entier



0 500 1.000 2.000 3.000 m  
Echelle non valable pour le secteur de Disneyland

Source et traitement des données : M. Bauder  
Fond de carte : OpenStreetMap 07/2013  
Projection : UTM Zone 31N

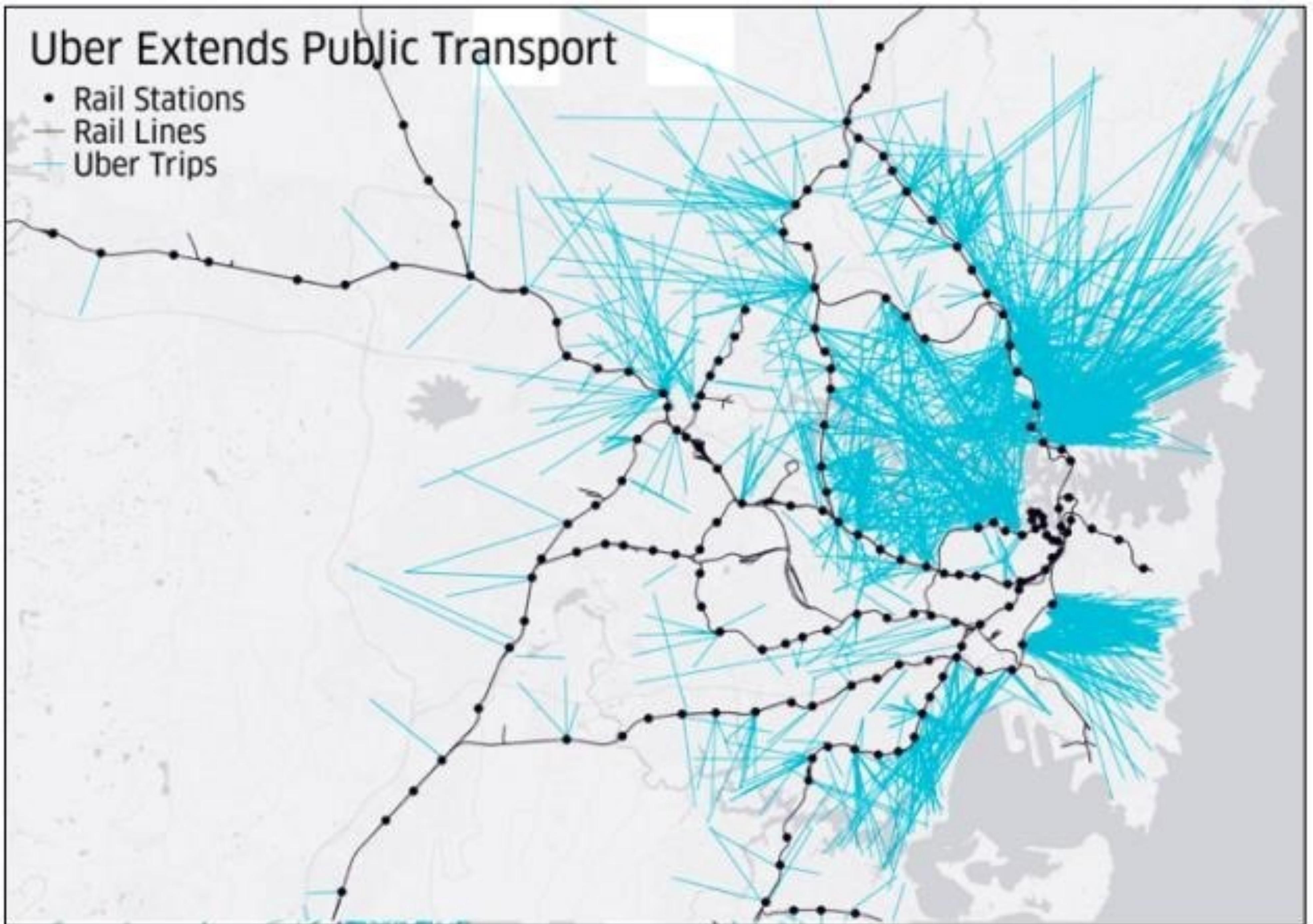
### Terrains d'étude :





# Uber Extends Public Transport

- Rail Stations
- Rail Lines
- Uber Trips



# Folium API + Quick Warm Up

# Folium



- Open source python library to visualize data on interactive leaflet map.
- Folium is a leaflet wrapper
- Exists since 2 years
- Last version 0.8.3

```
$ pip install folium
```

# Folium



- Main Folium API:
  - Map creation
  - Markers
  - Features
  - Layers
  - Plugins
  - Saving html maps
- Let's practice with the warm up notebook: exercice/Quick start with Folium API
- See tutorial: <https://python-visualization.github.io/folium/quickstart.html#Getting-Started>
- See modules: <https://python-visualization.github.io/folium/modules.html>

# **Hands on**

# Where to place NYC Taxi station ?

**Open Notebook:** exercice/ Hands\_on\_nyc\_taxi\_station

# New York City Taxi Trip Duration

Share code and data to improve ride time predictions

 Kaggle · 1,257 teams · a year ago

\$30,000

Prize Money

Overview   **Data**   Kernels   Discussion   Leaderboard   Rules   Team

My Submissions

Late Submission

## Data Description

### Data fields

- **id** - a unique identifier for each trip
- **vendor\_id** - a code indicating the provider associated with the trip record
- **pickup\_datetime** - date and time when the meter was engaged
- **dropoff\_datetime** - date and time when the meter was disengaged
- **passenger\_count** - the number of passengers in the vehicle (driver entered value)
- **pickup\_longitude** - the longitude where the meter was engaged
- **pickup\_latitude** - the latitude where the meter was engaged
- **dropoff\_longitude** - the longitude where the meter was disengaged
- **dropoff\_latitude** - the latitude where the meter was disengaged
- **store\_and\_fwd\_flag** - This flag indicates whether the trip record was held in vehicle memory before sending to the vendor because the vehicle did not have a connection to the server - Y=store and forward; N=not a store and forward trip
- **trip\_duration** - duration of the trip in seconds

We will be using the train data from the New York City Taxi Trip Duration Kaggle.

Link: <https://www.kaggle.com/c/nyc-taxi-trip-duration/data>

**To do:** Download 'train.csv' and put it in './**data\_used/**' folder

# Going further

# More tools

- Panads & GeoPanads
- Geopyspark
- For All tools: <https://github.com/sacridini/Awesome-Geospatial#python>

**Thanks !**